Off-Road Terrain Traversability Analysis and Hazard Avoidance for UGVs

Jacoby Larson, Mohan Trivedi, *Michael Bruch
Department of Electrical Engineering, University of California San Diego
{jrlarson, trivedi}@ucsd.edu
**SPAWAR Systems Center Pacific*
michael.bruch@navy.mil

***Abstract*—To achieve complete autonomy of unmanned ground vehicles (UGVs) in off-road terrain at high speeds, a robot must understand and analyze the terrain it is driving on in real-time just as a human analyzes the terrain and makes decisions of where to drive. Much research has been done in the way of obstacle avoidance, terrain classification, and path planning, but there has yet to be seen a system that can accurately traverse off-road environments at high speeds in an autonomous fashion. We present algorithms that analyze the off-road terrain using a point cloud produced by a 3D laser range finder, determine potential hazards both above ground and those where the ground cover has a negative slope, then plan safe routes around those hazards. We take into account problems such as high-centering, tip-over, and overhanging obstacles. In this paper we discuss a real-time approach to analyzing the traversability of off-road terrain for UGVs considering positive and negative obstacles using multi-level surface maps and planning paths to a goal state.**

***Index Terms*—UGV, lidar, velodyne, autonomous, autonomy, unmanned, off-road, point cloud, 3D, 2.5D, traversability, maps, path planning, multi-level surface, classification**

I. Introduction

Unmanned vehicle navigation and obstacle avoidance has had major breakthroughs in the last few years, showing that a vehicle can drive autonomously at high speeds in highly controlled desert and urban environments such as was proved in the Defense Advanced Research Projects Agency (DARPA) Grand Challenge and DARPA Urban Challenge [1]. As well, Jet Propulsion Laboratory (JPL) has shown that unmanned vehicles "Spirit" and "Opportunity" can navigate through the harshest of off-road environments, Mars. Yet when it comes to military operations in rugged off-road terrain, the technology is vastly behind. Explosive Ordnance Disposal technicians are still carefully maneuvering ground vehicles by remote control using the video feed on a bulky hardened laptop. This technology lag is in large part due to lack of real-time autonomous off-road traversability analysis for an unmanned ground vehicles (UGV). Military applications for UGVs such as resupply, casualty evacuation, surveillance, and reconnaissance must accommodate off-road terrain based upon the warfighting areas in which the US military is currently involved. Accurately representing off-road terrain and analyzing it in real-time is a challenge for most UGV robotic systems and the majority of UGVs operate at slow speeds over relatively flat terrain. Several recent Marine Corp Warfighting Laboratory (MCWL) limited objective experiments (LOEs) focused on the enhanced company operations scenario. These LOEs were conducted using commercially designed UGVs from companies such as GDRS, TORC, and others. The initial reports are that none of these systems can effectively navigate mountainous roads with moderate drops and tight turns. This research is aimed at developing methods of safely traversing rough terrain at high speeds by first detecting obstacles, second producing traversability scores for each terrain location, and finally planning a safe route to avoid those predetermined hazards.

II. Related Research

Due to the importance for robotic vehicle mobility, obstacle detection and avoidance for UGVs has been thoroughly explored in the past. The use of camera systems has received much more attention than the use of lidar. Passive ranging systems such as stereo cameras have proven to be beneficial to understanding the environment. JPL has demonstrated large gains in using passive systems such as stereo cameras [2, 3]: they have a low cost, do not emit electromagnetic signatures, are easy to fuse with color images, provide long range data, but oftentimes have a small angular resolution. On the other hand, lidar provides high resolution (even up to 360 degrees), produces higher quality range data, is getting cheaper, does not require as much computational time and hardware to return a point cloud, and was one of the main sensors used for those teams that successfully completed the DARPA challenges. Carnegie Mellon University has shown [4] how to use lidar to classify natural terrain into saliency features such as scatter, linear, or surface which can be used for traversability.

The point cloud data retrieved from the sensor systems can be modeled using a 2D, 2.5D, or 3D map. A 2D grid map, also known as an occupancy grid, uses the binary values of 1 or 0 as grid cells that hold obstacles or not. These systems are of great use for path planning and navigation since a robot is very interested in the location of obstacles. The 3D grid map is made up of voxels that take up considerably large amount of memory and complexity but are very useful for path planning in air and under water applications [5]. Because of the complexity and time required to analyze 3D data, a very common method of extraction is to model the terrain in a 2D grid with extended information and is referred to as 2.5D, which will hold much more meaningful information of the cell, more than just a binary value of a 2D occupancy map.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Off-Road Terrain Traversability Analysis and Hazard Avoidance for UGVs	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Department of Electrical Engineering,University of California San Diego, ,San Diego,CA**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

To achieve complete autonomy of unmanned ground vehicles (UGVs) in off-road terrain at high speeds, a robot must understand and analyze the terrain it is driving on in real-time just as a human analyzes the terrain and makes decisions of where to drive. Much research has been done in the way of obstacle avoidance, terrain classification, and path planning, but there has yet to be seen a system that can accurately traverse offroad environments at high speeds in an autonomous fashion. We present algorithms that analyze the off-road terrain using a point cloud produced by a 3D laser range finder, determine potential hazards both above ground and those where the ground cover has a negative slope, then plan safe routes around those hazards. We take into account problems such as high-centering, tip-over and overhanging obstacles. In this paper we discuss a real-time approach to analyzing the traversability of off-road terrain for UGVs considering positive and negative obstacles using multilevel surface maps and planning paths to a goal state.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Public Release**	**7**	

An accurate world map is not complete without a model of the surrounding obstacles. We refer to those hazards of negative slope such as a steep downhill, drop-off, or gap as a negative obstacle. These can be detected by looking for negative slopes that are too steep or gaps in data that exceed a width threshold[6, 7]. This analysis also needs to consider occlusions from other obstacles [8, 9]. As well, positive obstacles include anything standing out in the positive territory of the ground, such as rocks, trees, poles, buildings, and steep uphill slopes that might cause tip-over [10, 11]. In addition to detecting untraversable obstacles it is valuable to classify the traversability of each grid cell, making it restrictively harder (but not impossible) to plan paths through those areas.

Path Planning...

III. Approach

A. Terrain Representation & Multi-Level Surface Maps

To simplify the data and process it in real-time, we arrange the 3D laser points to fit inside a fixed size Cartesian grid of 2.5D cells, approximately 40 cm x 40 cm, measuring 100 meters in the x and y directions (see Figure 1). For reference, we use a right-handed coordinate system where x goes to the left and right of the vehicle, y measures what is in front and behind, and z comes out of the ground plane. Each cell in the 2.5D grid contains additional information, such as elevation, terrain traversability measures, and so on. Although this 2.5D grid can be very efficient for determining ground cover, it is limited in its ability to detect overhanging obstacles or pathways such as bridges or overhanging trees.

A multi-level surface map as discussed in [12, 13] allows multiple surface levels to be represented in one grid cell, which can aid in correctly classifying overhanging obstacles and provide additional search paths if one of those surfaces is a bridged road above the robot. The surfaces retain statistical information of all points that lie in it such as elevation mean and variance, number of points, and maximum and minimum elevation. The surface level has a variance threshold for the elevation variance of points that fit inside that cell and when exceeded, is referred to as a vertical surface or a positive obstacle. Otherwise the surface is referred to as a horizontal surface. Any point that lies above a surface level with a distance greater than the height of the robot from the maximum value of the lower surface level, will be considered a new surface level. Thus the traversability component of the robot may plan routes through grid cells with multiple layers. These surface levels are ordered from lowest to highest for fast lookup when adding new data points. If points are added that lie between two surface levels that remove the safe robot height distance between two surface levels, then those levels are merged into one and variances, means, and depth values are recalculated.

We represent the elevation of the data points that fall into the surface level as a Gaussian distribution. For fast processing and to reduce the amount of memory used, online calculations of mean and variance are used. In our case we used a formula for calculating an unbiased estimate of the population variance

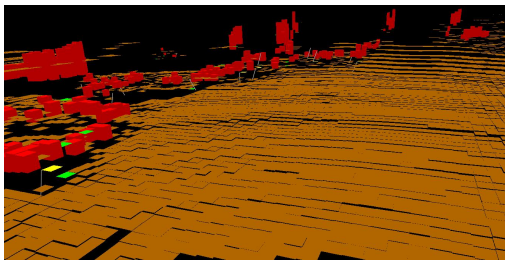


Figure 1: Grid cells calculated from the elevations of the data points of a single frame. (Brown color represents the ground while red, green, and yellow represents hazards)

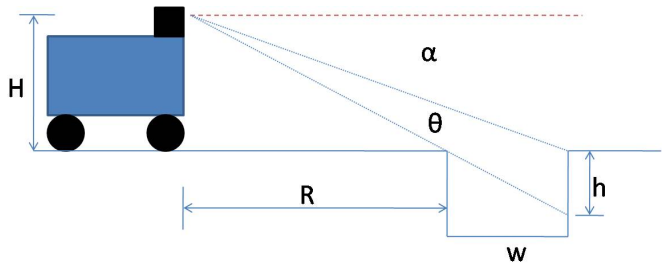


Figure 2: Geometry of negative obstacle detection

$$\sigma^2 = \frac{\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2/n}{n-1} \tag{1}$$

This calculation can easily be computed and merged with other surface levels by keeping track of the number of points that fall into this surface level, the cumulative sum of elevations, and cumulative sum of the squares of the elevations.

B. Negative Obstacles

Negative obstacles are ditches or terrain with a steep negative slope that if traversed would be a hazard to the vehicle. Negative obstacles can be just as hazardous to unmanned vehicles as obstacles above ground. In fact, negative obstacles are much harder to detect from close up and nearly impossible from far away. The equation for detecting negative obstacles comes from [14] and shown in Figure 2 where the width of the obstacle is w, H is the height of the sensor from the ground, h is the depth of the obstacle seen by the sensor, R is the range from the sensor to the obstacle. The equation to solve for θ is

$$\theta = \frac{Hw}{R(R+w)} \tag{2}$$

The angle θ decreases significantly as the range increases ($\frac{1}{R^2}$), which makes negative obstacles increasingly difficult to detect since the vertical angular resolution of the lidar also decreases with range. Yet detecting negative obstacles at greater ranges is essential because this application is intended for use on fast moving UGVs, upwards of 30mph, and faster

speeds requires a greater stopping distance. It is known from [15, 14] that the look-ahead distance needs to be at least:

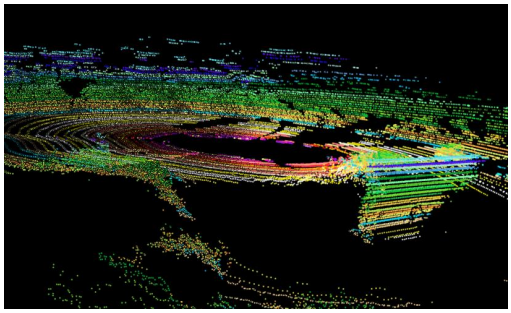
$$R = \frac{v^2}{2\mu g} + vT_r + B \tag{3}$$

where μ is the coefficient of static friction between ground and wheels with a common value of 0.65, g is gravitational acceleration with a value of 9.8, $T_r$is the total reaction time with a common value of 0.25, and B is a buffer distance used for safety with a value of 2 in our experiments. The velocity value becomes the dominant term at v>3.2 m/s: for a velocity of 15 mph or 6.7 m/s, the distance needed to stop is 7.2 m; for a velocity of 30 mph or 13.4 m/s, the distance needed to stop is 19.4 m.

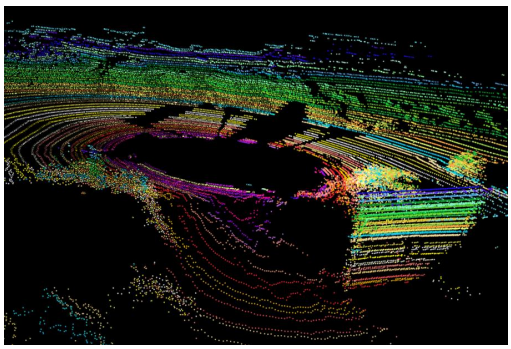
Because of the difficulty in detecting negative obstacles, our algorithm is very conservative and scans for any potential negative obstacles, then labels them for further analysis. We believe it is better to be cautious and safe when dealing with expensive robotic equipment. Figures 3a and 3b depict a large gap in data that would be tagged as a potential negative obstacle but when investigated further, turns out to be a steep slope that can be traversed.

We classify potential negative obstacles by first detecting gaps, patches of no data that exceed some width, where there could exist a ditch or negative slope. The search is done by tracing a ray of 3D points outward from the sensor, following the returns from the vertical alignment of lasers starting with the lowest angle of incidence towards the highest angle. If a gap is found with a distance greater than some width threshold (which will be variable dependent upon the vertical angle of incidence), then the data before it and after it is analyzed for clues. The width threshold value we used was greater than 1 meter as well as twice the distance of the width from the previous laser point to the beginning of the gap. It is important to determine if the gap was caused by a potential negative obstacle or a positive obstacle. A positive obstacle will have a prohibitively steep slope over a range of points prior to the gap. If the gap is a result of a positive obstacle, it will not be classified as a negative obstacle since the platform will not be able to traverse the positive obstacle anyway. If the gap is indeed classified as a negative obstacle, it is the result that either the negative elevation drop is significant (more than 1 meter for our purposes), or the data after the gap has a significant positive slope (as the sloping up-side of a ditch), see Figure 4.

Those sections of the ray tracings that are classified as potential negative obstacles (Figure 5a) are translated into its world model grid cell counterpart and recorded as a hit of a potential negative obstacle (Figure 5b) for purposes of traversability analysis. If enough hits are recorded in a cell, it is classified as a potential negative obstacle and displayed on the map. Because this algorithm is more cautious in classifying potential negative obstacles, it will report many false positives, especially at farther ranges, and should not be treated as untraversable. Instead the robot is allowed to approach these areas, but should do so with caution and reduced speed, which can be implemented into the path planning cycle. This software will be integrated with SPAWAR Systems Center Pacific's Autonomous Capabilities Suite (ACS) architecture. ACS could use it's fuzzy logic planner [16] to place potential negative obstacles as obstacles in the range abstraction regions *l_front* or *r_front*, giving them a fuzzy set value of *Close* or *Not Close*, which according to the fuzzy associate memory rules, causes the robot to approach more slowly but not stop.



(a)



(b)

Figure 3: 3D point cloud returns from sequential frames. The area in the bottom middle of each image has a steep negative slope. In (3a) the slope hasn't been detected yet and would be classified as a potential negative obstacle. In (3b) there is more data and the area can now be correctly detected as a negative slope.

Those potential negative obstacles that are within a short range of the robot should be considered actual negative obstacles and avoided.

C. Traversability Analysis

A cell receives a traversability score based on the existence of a positive obstacle, a step edge obstacle, the steepness of the slope of its surrounding neighbors, and the slope residual (roughness). Positive obstacle detection was already performed when the data were inserted into the multi-level surface map. A positive obstacle is revealed when the elevation variance of a surface level (calculated when the 3D data points fall into the surface level of a cell) exceeds a variance threshold: this method ignores extreme outliers when there is enough data. Another hazard feature called a step edge can be detected by

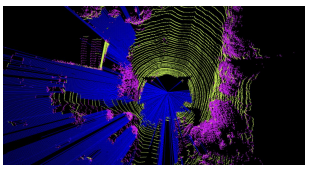
(a) Gap followed by a large drop in elevation: potential negative obstacle

(b) Gap followed by a steep uphill slope: potential negative obstacle

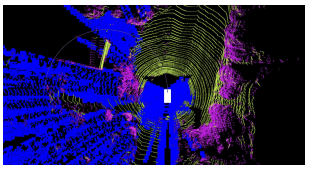
(c) Slight uphill slope followed by a gap and drop in elevation: potential negative obstacle

(d) Positive obstacle followed by a gap: not a potential negative obstacle

Figure 4: Ray tracing examples with results determined during potential negative obstacle detection step



(a) Potential negative obstacle rays



(b) Potential negative obstacle pixels

Figure 5: Potential negative obstacle displays

calculating the elevation differences between cells, which is UGV size dependent. If the UGV is small with small tires or treads, then even a small gap in elevation could block or tip over the robot. In addition, the slope of the surrounding area of each cell is calculated by fitting the points to a plane and finding the surface normal of that plane using the eigenvector of the smallest eigenvalue of the covariance matrix. At this point in the analysis, all the of individual data point information has been erased and only the statistical information for each surface level of the cells remain. To calculate the covariance matrix, we will be using the mean of the elevation (the z component) and the x and y values of each of the grid cells, centered from the robot. Note that it is not required to calculate the slope of the obstacle cells since the vehicle will not be traveling over them. In our tests we use the 8 connected neighbors of each cell, and only calculate the slope if there are at least 4 neighbors. The $\bar{x}$, $\bar{y}$, and $\bar{z}$ represent the mean values of the x, y, and z values of the neighbors.

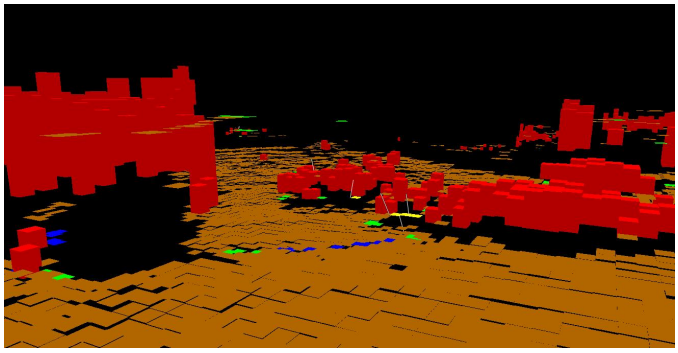


Figure 6: Grid cells painted on the screen with various colors representing the its classification: brown is a horizontal cell (ground), blue is a potential negative obstacle, green is a step edge, yellow is slope (with surface normal included), and red is a positive obstacle

$$M = \begin{bmatrix} (x_1 - \bar{x}) & (y_1 - \bar{y}) & (z_1 - \bar{z}) \\ (x_2 - \bar{x}) & (y_2 - \bar{y}) & (z_2 - \bar{z}) \\ \dots & \dots & \dots \\ (x_n - \bar{x}) & (y_n - \bar{y}) & (z_n - \bar{z}) \end{bmatrix} \tag{4}$$

Lets define $A = M^T M$, which when divided by the number of data points becomes the covariance matrix of the data. To find the eigenvalues and eigenvectors of the covariance matrix, we use the singular value decomposition (SVD). But we really only need M, not A. $M = USV^T$where U is an orthogonal matrix, S is a diagonal matrix of the singular values of M, and the columns of V contain its singular vectors.

$$A = M^T M = (USV^T)^T(USV^T) = VS^T U^T USV^T = VS^2V^T \tag{5}$$

This shows us that the singular values of A are just the squares of the singular values of M and the singular vectors of A are the singular values of M.

If the absolute value of the surface normal difference from the gravity normal is too great, then it is classified as a slope obstacle. We also use the residual of the plane fit function to determine roughness of the area, which is added into the traversability of the cell.

After the cell has been given its traversability score based on obstacles, step edge, slope, and slope residual (see Figure 6), it is time to choose the correct path.

D. Path Planning

The data collected from most lidar sensors, mounted on a robotic vehicle, without any a priori information, will not provide adequate range to create a large enough map to plan optimal paths to a goal location of any significant distance. Our methods focus on the near field path planning using an arc-based planner (Figure 7), similar to those done in [17] as well as on the Mars Rovers. The path planning algorithm simulates placing the platform centered on the cells of the arc (starting at the sensor and moving outward) and summing the traversability scores for all the cells the vehicle would

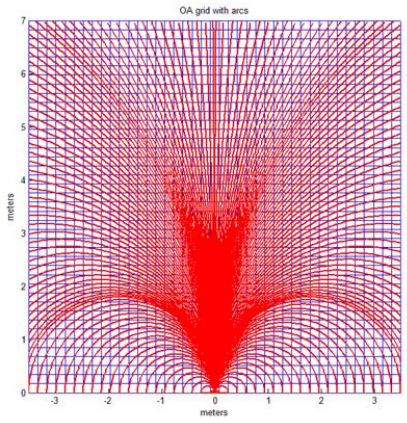


Figure 7: Reactive path planning arcs

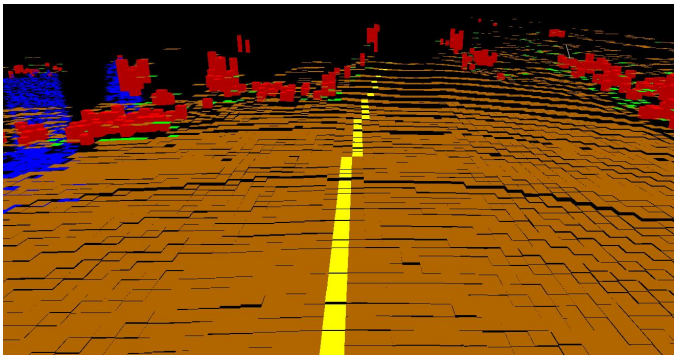


Figure 8: Chosen arc

run over (this usually includes a wider selection of cells than just those just in the arc). The traversability score for each cell should already be calculated based on the terrain found inside it: positive obstacles, step edges, slope, and slope residual. Furthermore, penalties are added for tip-over and high-centering, which can be calculated from all of the cells underneath the vehicle. Due to the inherent curvature of the arcs and the disconnect between the actual ackerman steering capabilities of the vehicle (rotating in place), there will be arcs chosen which provide the best route for the first section of the path but eventually curve into obstacles. Since the robot will be calculating the traversability arc multiple times a second and revising its arc selection just as quickly, more weight is given to the cells of the arc in the closer regions than those in the farther regions. Deviation from the desired arc (the predetermined path) is also added into the score. As each individual pixel score is being accumulated, if the total score exceeds a threshold of acceptability, the process is halted for this arc and the arc is thrown away. The arc with the best score is kept (See Figure 8). For additional speed and reduced search, once an arc with a "good enough" score is found, it can be chosen as the desired arc and the search ends.

Figure 9: Max ATV platform

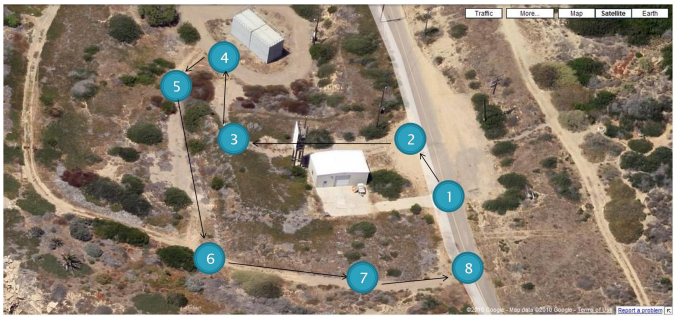
E. Platform & Sensors

All of the traversability analysis methods and software we have developed have been exclusively for use on an unmanned ground vehicle with a 3D lidar sensor. Our first intended platform is a Max ATV, Figure 9, mounted with a roll bar to protect against tipping. The vehicle primarily depends on the Velodyne HDL-64E sensor for data of the surrounding environment, a lidar system with 64 lasers delivering 360 degree horizontal field of view with 26.8 degree vertical field of view, providing 100,000 data points at 10Hz. This system provides readings of range and intensity out to a distance of 120 meters. We plan to implement a vertical tilting mechanism on the lidar sensor to get a better view of negative obstacles at close ranges. The goal is to implement the traversability analysis software to be UGV platform independent and we have already begun testing this software on data sets captured from a Hokuyo UTM-30LX lidar sensor, set in a vertically rotating mechanism, mounted on an iRobot Packbot, with good results.

IV. Results

One of our main objectives with this project has been to provide a system that can not only detect hazards and avoid them, but to do it in real-time, that is to say, allow the robot to make decisions about where to go while it is going. The Max ATV platform has yet to be fully developed to be controlled by software, so all of our results have been performed on real data collected from the lidar sensor mounted on the vehicle which is manually driven over an off-road course. An image of the course can be seen in Figure 10. The lidar collected 1122 full frames of point cloud data, and our algorithm was able to process each frame, detecting all hazards and providing a suggested route, as well as display the data on the screen, at an average rate of 2.23 Hz. on a dual-core laptop machine. We believe this rate to be sufficient for a vehicle to perform hazard detection and avoidance at speeds of up to 10 mph (4.5 m/s), as long as the hazards can be detected at sufficient ranges. The ranges of hazard detection in this data set are provided in table I.

Figure 10: Off-road course Google sky-view image



Hazard Feature	Max. Detection Range	Description
Positive obstacle	113.4	Vegetation
Negative obstacle	82.6	Steep slope
Overhang	100.6	Tall vegetation
Step edge	115.1	Steep hill
Steep slope	115.1	Steep hill

Table I: Hazard detection ranges

V. Future Work

We have noticed that even in off-road environments, there is usually some worn down pathway that resembles a road, and that a UGV will have a much easier time navigating if it can stay on that road. We plan to add a scoring weight into the path planning step to favor the trajectories that stay on the road.

As well, in off-road terrain, many paths have overgrown vegetation that, to a simple obstacle detector, appear to be an obstacle. Thus the traversability analysis will block those paths even if a human operator would easily drive through the path and let a few branches scrape the vehicle. We have run a simple method of a sliding window over horizontal sections of points, looking for straight lines, with the idea that vegetation will not have straight lines. For a majority of the vegetation this method classifies correctly (see Figure 11), but has a high false positive rate, especially for thin structures, like metal poles or thin tree trunks. One of the ways we plan on classifying vegetation and for further traversability analysis of that vegetation is to cluster the data points by location as well as add a calibrated video camera for further color segmentation and run principle component analysis (PCA) to classify scatter, linear, or planar objects.

Due of the limitation of the close-range sensors to create a large global map to be used for path planning to a goal location, we shall be researching the best ways to use aerial imagery, fused with local obstacle data obtained from our sensors, to obtain a general idea of the direction to travel to reach a goal.

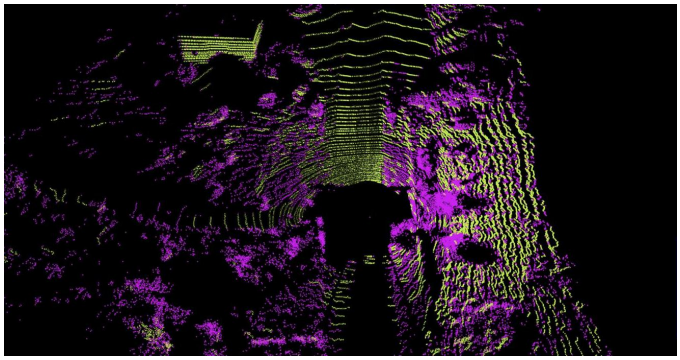


Figure 11: Scene showing vegetation classification example (yellow is smooth surface and purple is the vegetation).

References

[1] Sebastian Thun et al. Stanley: The robot that won the darpa grand challenge: Research articles. *Journal of Field Robotic*, v.23 n.9:661–692, September 2006.

[2] A. Rankin, A. Huertas, and L. Matthies. Evaluation of stereo vision obstacle detection algorithms for off-road autonomous navigation. In *AUVSI Symposium on Unmanned Systems*, 2005.

[3] A. Rankin, A. Huertas, and L. Matthies. Stereo vision based terrain mapping for off-road autonomous navigation. In *Proceedings of SPIE*, volume 7332, Orlando, FL, 2009.

[4] J-F. Lalonde, N. Vandapel, D. Huber, and M. Hebert. Natural terrain classification using thre-dimensional ladar data for ground robot mobility. *Journal of Field Robotics*, 23(10):839–861, November 2006.

[5] J. Carsten, D. Ferguson, and A. Stentz. 3d field d*: Improved path planning and replanning in three dimensions. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3381–3386, October 2006.

[6] H. Seraji. Rule-based traversability indices for multi-scale terrain asessment. In *Proceedings of 2003 IEEE International Conference on Control Applications*, Istanbul, Turkey, June 2003.

[7] A. Murarka, M. Sridharan, and B. Kuipers. Detecting obstacles and drop-offs using stereo and motion cues for safe local motion. In *International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[8] T. Hong, S. Legowik, and M. Nashman. Obstacle detection and mapping system. *National Institute of Standards and Technology (NIST) Technical Report NISTIR 6213*, pages 1–22, 1998.

[9] N. Heckman, J-F. Lalonde, N. Vandapel, , and M. Hebert. Potential negative obstacle detection by occlusion labeling. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[10] P. Roan, A. Burmeister, A. Rahimi, K. Holz, and D. Hooper. Real-world validation of three tipover algorithms for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, May 2010.

[11] S. A. A. Moosavian and K. Alipour. Stability evaluation of mobile robotic systems using moment-height measure. In *Proceedings of the IEEE International Conference on Robotics, Automation, and Mechatronics*, pages 97–102,

June 2006.

[12] R. Triebel, P. Pfaff, and W. Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *Proceedings of the International Conference on Intelligent Robots and Systems, IROS*, 2006.

[13] Dominik Joho, Cyrill Stachniss, Patrick Pfaff, and Wolfram Burgard. Autonomous exploration for 3d map learning. *Autonome Mobile Systeme (AMS), Kaiserslautern, Germany*, 2007.

[14] L. Matthies and A. Rankin. Negative obstacle detection by thermal signature. In *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 906–913, 2003.

[15] L. Matthies and P. Grandjean. Stochastic performance modeling and evaluation of obstacle detectability with imaging range sensors. *IEEE Transactions on Robotics and Automation*, 16(12), December 1994.

[16] B. Sights, G. Ahuja, G. Kogut, E.B. Pacis, H.R. Everett, D. Fellars, and S. Hardjadinata. Modular robotic intelligence system based on fuzzy reasoning and state machine sequencing. In *SPIE Proceedings 6561: Unmanned Systems Technology IX, Defense and Security Symposium*, Orlando, FL, April 2007.

[17] J. Larson, M. Bruch, and J. Ebken. Autonomous navigation and obstacle avoidance for unmanned surface vehicles. In *Proceedings of SPIE Unmanned Systems Technology VIII*, Orlando, FL, April 2006.